

# Personal Assistant

## Introducere

Personal assistant este un dashboard inteligent care poate comunica cu dispozitivele tale pentru a afișa informații despre calendar, remindere, vreme, oră, notificări, temperatură / umiditatea din cameră etc. Se poate conecta la telefon pentru a trimite notificări și are un buzzer pentru alarme / remindere. De asemenea, folosește un senzor de lumină pentru a te trezi odată cu răsăritul.

Scopul proiectului este de a avea acces ușor la informațiile utile ție. Proiectul este gândit să stea pe birou și să afișeze diverse lucruri, precum: valori de la senzori, informații de la telefon, informații de la un API etc. sau să te notifice de ceva prin intermediul buzzer-ului.

Inspirația pentru proiectul meu a venit în momentul în care am cumpărat un "Sunrise Alarm Clock" de pe Temu care este configurabil printr-o aplicație de telefon. Mi s-a părut super mișto ideea de a controla un dispozitiv prin internet (nu mi-a plăcut niciodată să setez un ceas cu butoane) și m-am gândit să fac și eu ceva asemănător, dar mult mai configurabil. Am vrut totuși să păstrez funcționalitate de sunrise alarm :)

Cred că un astfel de dashboard este util pentru oricine, deoarece poate fi configurat să afișeze aproape orice folosind un ESP32.

## Descriere generală

Proiectul are la bază un ESP32 și un display LCD. Am decis să folosesc un ESP32, deoarece am vrut conexiune la Internet și Bluetooth pentru a putea comunica cu telefonul / un server. Pe lângă aceste dispozitive, proiectul mai conține un buzzer pentru alerte sonore și doi senzori: un senzor pentru temperatură / umiditate și un senzor de lumină pentru detecția răsăritului.



Modul de funcționare este următorul:

- ESP32-ul cere informații din diverse surse: valori de la senzori, notificări de la telefon, lista cu remindere de la un server etc.
- Informațiile primite sunt prelucrate (datele sunt extrase dintr-un pachet Ethernet, valorile de la senzori sunt interpretate etc.)
- Microcontroller-ul actualizează UI-ul și trimite datele la display **sau** acționează buzzer-ul

## Hardware Design

Listă de piese:

- 1 x Placă ESP-WROOM-32
- 1 x Display TFT SPI 2.8 inch 240×320 LCD cu touchscreen
- 1 x Senzor digital de lumina BH1750
- 1 x Senzor de Temperatura si Umiditate DHT22
- 1 x Buzzer Pasiv
- Fire de legătură
- 1 x Breadboard 830 puncte

Aici puneți tot ce ține de hardware design:

- listă de piese
- scheme electrice (se pot lua și de pe Internet și din datasheet-uri, e.g. <http://www.captain.at/electronic-atmega16-mmc-schematic.png>)
- diagrame de semnal
- rezultatele simulării

## Software Design

Descrierea codului aplicației (firmware):


- mediu de dezvoltare (if any) (e.g. AVR Studio, CodeVisionAVR)
- librării și surse 3rd-party (e.g. Procyon AVRlib)
- algoritmi și structuri pe care plănuiți să le implementați
- (etapa 3) surse și funcții implementate

## Rezultate Obținute

Care au fost rezultatele obținute în urma realizării proiectului vostru.

## Concluzii

## Download

O arhivă (sau mai multe dacă este cazul) cu fișierele obținute în urma realizării proiectului: surse, scheme, etc. Un fișier README, un ChangeLog, un script de compilare și copiere automată pe uC crează întotdeauna o impresie bună .

Fișierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fișierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume\_student** (dacă este cazul).  
**Exemplu:** Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru\_alin**.

## Jurnal

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

## Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** și **Resurse Hardware**.

[Export to PDF](#)

From:  
<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:  
<http://ocw.cs.pub.ro/courses/pm/prj2026/ciprian.popescu0411/alexandru.nehoianu> 

Last update: **2026/05/09 18:30**